

## IN THE SPECIFICATION

Please amend the specification as follows:

1. At page 2, lines 9-19:

One of the technologies underlying the World Wide Web is the prospect of using component software technology -- the idea of breaking large, complex software applications into a series of pre-built and easily developed, understood, and changed software modules called components -- as a means to deliver software solutions much more quickly and at a lower cost ~~(source: DCOM: A Business Overview, online at <http://www.microsoft.com/ntserver/guide/dcom.asp>)~~. The goal is to achieve economies of scale for software deployment across the industry.

2. At page 19:

**Fig 3** is a functional block diagram of a populated embodiment of the present invention showing a plurality of clients, adapters, and connectors. Client 0<sub>L</sub> 92<sub>L</sub> prepares service requests in a first format, such as ASP. Client 1<sub>L</sub> 90<sub>L</sub> prepares service requests in a second format, such as visual basic. Client m<sub>L</sub> 94<sub>L</sub> prepares service requests in a third format.

Each of the respective clients sends its service requests to the respective associated adapter as shown. Client 0<sub>L</sub> 92<sub>L</sub> service requests are sent to Adapter 0<sub>L</sub> 84. Client 1<sub>L</sub> 90<sub>L</sub> service requests are sent to Adapter 1<sub>L</sub> 86, and Client m<sub>L</sub> 94<sub>L</sub> service requests are sent to Adapter m<sub>L</sub> 88. As explained above, each of these adapters (i.e., 84, 86, and 88) converts the inputs to a single format for input to GGATE 72.

The operation of GGATE 72 is explained in detail in the above cross-referenced U.S. Patent Applications. The converted and process service requests are transferred to the appropriate one of Connector 0<sub>L</sub> 73, Connector 1<sub>L</sub> 74, or Connector m<sub>L</sub> 75. These connectors are arranged to handle the different formats as with the plurality of adapters. The converted, processed, and connected service requests are transferred to End Service Provider 78 for honoring.

3. At page 3, line 19, through page 4, line 4:

DCOM has its roots in Microsoft's object technology, which has evolved over the last decade from DDE (Dynamic Data Exchange, a form of messaging between Windows programs), OLE (Object Linking and Embedding, embedding visual links between programs within an application), COM (the Component Object Model, used as the basis for all object binding), and ActiveX (COM enabled for the Internet) objects. As stated earlier, applications built from components are simply easier to debug and evolve than large, monolithic applications.

4. At page 17, lines 1-7:

**FIG. 2** is a functional diagram of the major components of the preferred mode of the present invention. In this example, DCOM client 68 generates a service request using ~~visual basic~~ Visual Basic language (i.e., VB), C++, active server page (ASP), or other suitable tools. The service request is transferred to DCOM Server (adapter) 70. As explained in detail below, there is a different adapter for each input format employed.

5. At page 18, lines 1-7:

**Fig 3** is a functional block diagram of a populated embodiment of the present invention showing a plurality of clients, adapters, and connectors. Client 0 92 prepares service requests in a first format, such as ASP. Client 1 90 prepares service requests in a second format, such as ~~visual basic~~ Visual Basic language. Client m 94 prepares service requests in a third format.

**THIS PAGE BLANK (USPTO)**